Attorney Docket No. 042390P11921
Express Mail No. EL 863955774 US

099778.101501

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR AN ATOMIC OPERATION IN A PARALLEL COMPUTING ENVIRONMENT

Inventors:

David K. Poulsen
Sanjiv M. Shah
Paul M. Petersen
Grant E. Haab

Prepared by:

Blakely, Sokoloff, Taylor & Zafman LLP
12400 Wilshire Blvd., Suite 700
Los Angeles, California 90025
(310)207-3800

"Express Mail" mailing label number: EL 863955774 US Date of Deposit: October 15, 2001

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner of Patents, Washington, D. C. 20231.

Shenise Ramdeen
(Typed or printed name of person mailing paper or fee)

Shenise Ramdeen
(Signature of person mailing paper or fee)

October 15, 2001
(Date signed)

Method and Apparatus for an Atomic Operation in a Parallel Computing Environment

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The invention relates to the field of computer processing. More specifically, the invention relates to a method and apparatus of parallel computation.

Background of the Invention

[0002] Parallel computing of tasks achieves faster execution and/or enables the performance of complex tasks that single processor systems cannot perform. One paradigm for performing parallel computing is shared-memory programming. The OpenMP standard is an agreed upon industry standard for programming shared-memory architectures.

[0003] OpenMP provides for synchronization constructs. One of these constructs ensures that memory locations are updated atomically: the ATOMIC construct. An atomic update operation is a sequence of operations or instructions that are non-interruptible to update a memory location. Atomically updating a shared memory location prevents multiple threads of a team from performing the same operation and/or destroying work done by another thread.

[0004] Although OpenMP outlines requirements for constructs and provides guidelines for parallel programming, details for implementing the ATOMIC construct are not provided. One method for atomically updating a shared memory location is to acquire locks on the memory location in order to limit modification of the shared memory location to the lock holder. Although this method ensures atomic updating of the memory location, the updating thread reduces performance with the acquisition and release of locks on the memory location. In addition, performance is reduced since other threads of the updating thread's team must wait to update the memory location until the lock on the memory location is released.

[0005] Another method to atomically update a memory location that achieves optimal performance is to create platform specific low-level instructions to perform the update. Although vendors can optimize their system with such low-level instructions, the cost to produce the low-level instructions can become a combinatorial explosion. To support such an implementation, vendors would create a number of low-level instructions proportional to the product of the number of data-types, the number of sizes of data-types, and the number of operations to be supported by a single platform. Hence, the cost of creating low-level instructions to support atomic updates for numerous operations is prohibitive on a single platform. This prohibitive cost makes a multiple-platform implementation of atomic update operations, based solely on platform specific low-level instructions, infeasible.